ANNEXURE-II

❖ SOURCE CODE FOR FREQUENCY MEASUREMENT

```
%PROGRAM FOR FREQUENCY MEASUREMENT USING MODIFIED SEZI MODEL
%DATE: 03/08/2007

function frequency = cal_freq(inpf,inporder,inpmag)

% bandpass filter coefficients
band_pass_coeff=[-0.00016
-0.001014
-0.001929
-0.002577
-0.002298
-0.000429
0.003065
0.00705
0.009242
0.007026
-0.001079
-0.013945
-0.027177
-0.033882
-0.02689
-0.00173
0.040841
0.093951
0.146003
0.18405
0.198
0.18405
0.146003
0.093951
0.040841
-0.00173
-0.02689
-0.033882
-0.027177
-0.013945
-0.001079
0.007026
0.009242
0.00705
0.003065
-0.000429
-0.002298
-0.002577
-0.001929
-0.001014
-0.00016];
```

```
% Low pass filter coefficients
low_pass_coeff = [
                0.5
                0
                0
                0
                0
                1
                0
                0
                0
                0
                0.5];
% ALL pass filter coefficients
all_pass_coeff = [
                0
                0
                0
                0
                0
                1
                0
                0
                0
                0
                0];

% General Parameters
ts = 1e-3;
%handles = guihandles(popupmenu9)
%f = guidata(handles,selected_string);
f = inpf;
Vm = 1;
V3h = 0.2;
V5h = 0;
V7h = 0;
Vh = 0.001;
pi = 3.14159265;
t = 0 : 0.001:0.12;

%Sine wave equation
num_of_samples = 120

if (HARMONIC_MODE == ON)
   v = Vm*sin(2*pi*f*t)+inpmag*sin(2*pi*inporder*f*t)
end
if (ONE_THIRD_HARMONIC == ON)
    v = Vm*sin(2*pi*f*t)+ Vh*sin(2*pi*(1/3)*f*t)
end
if (NORMAL_MODE == ON)
    v = Vm*sin(2*pi*f*t)
end

if (inporder == 0.3)
        inpmag =inpmag/10000;
end
```

```
v = Vm*sin(2*pi*f*t)+inpmag*sin(2*pi*inporder*f*t)
%plot(t,v);

%Bandpass filter implementation
num_of_bp_samples = 80;
bp_coeff_len = 41;
temp = 0;
total_bp_len = num_of_samples - bp_coeff_len;
for count = 1:1:total_bp_len
    temp1 = (bp_coeff_len+count);
    for i = 1:1:bp_coeff_len
         temp = band_pass_coeff1(i)*v(temp1-i) + temp;
    end
    bp_output(count) = temp;
    temp = 0;
end

%Lowpass filter implementation
lp_coeff_len = 11;
temp = 0;
total_lp_len = bp_coeff_len;
for count = 1:1:total_lp_len
    temp1 = (lp_coeff_len + count);
    for i = 1:1:lp_coeff_len
         temp = low_pass_coeff(i)*bp_output(temp1-i) + temp;
    end
    lp_output(count) = temp;
    temp = 0;
end

%Allpass filter implementation
ap_coeff_len = 11;
temp = 0;
total_ap_len = bp_coeff_len;
for count = 1:1:total_ap_len
    temp1 = (ap_coeff_len + count);
    for i = 1:1:ap_coeff_len
         temp = all_pass_coeff(i)*bp_output(temp1-i) + temp;
    end
    ap_output(count) = temp;
    temp = 0;
end

%IIR filter implementation after LPF and APF
temp = 0;
iir_lpf_output(1) = 0.1*(lp_output(lp_coeff_len +
1)*lp_output(lp_coeff_len + 1) - lp_output(1)*lp_output(1));
iir_apf_output(1) = 0.1*(ap_output(ap_coeff_len +
1)*ap_output(ap_coeff_len + 1) - ap_output(1)*ap_output(1));
div_output(1)= iir_lpf_output(1)/iir_apf_output(1);
sq_output(1) = sqrt(div_output(1));
temp2(1) = (1/5)*acos(sq_output(1) - 1);
freq(1) = (temp2(1) * 1/ts)/(2*pi);
```

```
for count = 2:1:21
    iir_lpf_output(count) = 0.1*((lp_output(lp_coeff_len +
count)*lp_output(lp_coeff_len + count))-
(lp_output(count)*lp_output(count))) + iir_lpf_output(count -1);
    iir_apf_output(count) = 0.1*((ap_output(ap_coeff_len +
count)*ap_output(ap_coeff_len + count))-
(ap_output(count)*ap_output(count))) + iir_apf_output(count -1);
    div_output(count) = iir_lpf_output(count)/iir_apf_output(count);
    sq_output(count) = sqrt(div_output(count));
    temp2(count) = (1/5)*acos(sq_output(count) - 1);
    freq(count) = (temp2(count) * 1/ts)/(2*pi);
end

%Frequency calculation block
sum = 0;
for count = 1:1:21
    sum = freq(count) + sum;
end
frequency = sum/21;
```

❖ SOURCE CODE FOR POWER MEASUREMENT

```
 function active_power = cal_power()


r_phase = (0 * pi)/180;
y_phase = (240 * pi)/180;
b_phase = (120 * pi)/180;

Vr = Vamp*sin((2*pi*freq*t) + r_phase);
Vy = Vamp*sin((2*pi*freq*t) + y_phase);
Vb = Vamp*sin((2*pi*freq*t) + b_phase);

Ir = Iamp*sin((2*pi*freq*t) + r_phase + phase_shift);
Iy = Iamp*sin((2*pi*freq*t) + y_phase + phase_shift);
Ib = Iamp*sin((2*pi*freq*t) + b_phase + phase_shift);

subplot(3,4,1);
plot(t,Vr,'b-',t,Ir,'r-');
xlabel('t');
ylabel('V');
title('PF = 0.8 lead');

Valpha = 0.816*Vr - 0.41*Vy - 0.41*Vb;
Vbeta  = 0.707*Vy - 0.707*Vb;
Vzero  = 0.577*Vr + 0.577*Vy + 0.577*Vb;

Ialpha = 0.816*Ir - 0.41*Iy - 0.41*Ib;
Ibeta  = 0.707*Iy - 0.707*Ib;
Izero  = 0.577*Ir + 0.577*Iy + 0.577*Ib;

active_power = Valpha.*Ialpha + Vbeta.*Ibeta;
reactive_power = Valpha.*Ibeta - Vbeta.*Ialpha;

subplot(3,4,5)
plot(t,active_power,t,reactive_power);
xlabel('t');
ylabel('P');

value_act_p = mean(active_power);
value_act_q = mean(reactive_power);

%First quadrant
if(value_act_p > 0 & value_act_q > 0)
    angle_p = 30;
 end

%Second quadrant
if(value_act_p < 0 & value_act_q > 0)
    angle_p = 120;
 end
```

```
%Third quadrant
if(value_act_p < 0 & value_act_q < 0)
    angle_p = 210;
 end

%Fourth quadrant
if(value_act_p > 0 & value_act_q < 0)
    angle_p = 310;
end

value = abs(value_act_p);
temp = (angle_p*pi)/180;
[x,y] = pol2cart(temp,value);
subplot(3,4,9);
compass(x,y,'r-');

%second case
angle = 120;
phase_shift = (angle*pi)/180;

Ir = Iamp*sin((2*pi*freq*t) + r_phase + phase_shift);
Iy = Iamp*sin((2*pi*freq*t) + y_phase + phase_shift);
Ib = Iamp*sin((2*pi*freq*t) + b_phase + phase_shift);

subplot(3,4,2);
plot(t,Vr,'b-',t,Ir,'r-');
xlabel('t');
ylabel('V');
title('PF = -0.5 lag');

Valpha = 0.816*Vr - 0.41*Vy - 0.41*Vb;
Vbeta  = 0.707*Vy - 0.707*Vb;
Vzero  = 0.577*Vr + 0.577*Vy + 0.577*Vb;

Ialpha = 0.816*Ir - 0.41*Iy - 0.41*Ib;
Ibeta  = 0.707*Iy - 0.707*Ib;
Izero  = 0.577*Ir + 0.577*Iy + 0.577*Ib;

active_power = Valpha.*Ialpha + Vbeta.*Ibeta;
reactive_power = Valpha.*Ibeta - Vbeta.*Ialpha;

subplot(3,4,6)
plot(t,active_power,t,reactive_power);
xlabel('t');
ylabel('P');

value_act_p = mean(active_power);
value_act_q = mean(reactive_power);

%First quadrant
if(value_act_p > 0 & value_act_q > 0)
    angle_p = 30;
 end
```

```
%Second quadrant
if(value_act_p < 0 & value_act_q > 0)
    angle_p = 120;
 end

%Third quadrant
if(value_act_p < 0 & value_act_q < 0)
    angle_p = 210;
 end

%Fourth quadrant
if(value_act_p > 0 & value_act_q < 0)
    angle_p = 310;
end

value = abs(value_act_p);
temp = (angle_p*pi)/180;
[x,y] = pol2cart(temp,value);
subplot(3,4,10);
compass(x,y,'r-');

%third case
angle = 210;
phase_shift = (angle*pi)/180;

Ir = Iamp*sin((2*pi*freq*t) + r_phase + phase_shift);
Iy = Iamp*sin((2*pi*freq*t) + y_phase + phase_shift);
Ib = Iamp*sin((2*pi*freq*t) + b_phase + phase_shift);

subplot(3,4,3);
plot(t,Vr,'b-',t,Ir,'r-');
xlabel('t');
ylabel('V');
title('PF = -0.8 lag');

Valpha = 0.816*Vr - 0.41*Vy - 0.41*Vb;
Vbeta  = 0.707*Vy - 0.707*Vb;
Vzero  = 0.577*Vr + 0.577*Vy + 0.577*Vb;

Ialpha = 0.816*Ir - 0.41*Iy - 0.41*Ib;
Ibeta  = 0.707*Iy - 0.707*Ib;
Izero  = 0.577*Ir + 0.577*Iy + 0.577*Ib;

active_power = Valpha.*Ialpha + Vbeta.*Ibeta;
reactive_power = Valpha.*Ibeta - Vbeta.*Ialpha;

subplot(3,4,7)
plot(t,active_power,t,reactive_power);
xlabel('t');
ylabel('P');

value_act_p = mean(active_power);
value_act_q = mean(reactive_power);
```

```
%First quadrant
if(value_act_p > 0 & value_act_q > 0)
    angle_p = 30;
 end

%Second quadrant
if(value_act_p < 0 & value_act_q > 0)
    angle_p = 120;
 end

%Third quadrant
if(value_act_p < 0 & value_act_q < 0)
    angle_p = 210;
 end

%Fourth quadrant
if(value_act_p > 0 & value_act_q < 0)
    angle_p = 310;
end

value = abs(value_act_p);
temp = (angle_p*pi)/180;
[x,y] = pol2cart(temp,value);
subplot(3,4,11);
compass(x,y,'r-');

%fourth case
angle = 310;
phase_shift = (angle*pi)/180;

Ir = Iamp*sin((2*pi*freq*t) + r_phase + phase_shift);
Iy = Iamp*sin((2*pi*freq*t) + y_phase + phase_shift);
Ib = Iamp*sin((2*pi*freq*t) + b_phase + phase_shift);

subplot(3,4,4);
plot(t,Vr,'b-',t,Ir,'r-');
xlabel('t');
ylabel('V');
title('PF = 0.6 lead');

Valpha = 0.816*Vr - 0.41*Vy - 0.41*Vb;
Vbeta  = 0.707*Vy - 0.707*Vb;
Vzero  = 0.577*Vr + 0.577*Vy + 0.577*Vb;

Ialpha = 0.816*Ir - 0.41*Iy - 0.41*Ib;
Ibeta  = 0.707*Iy - 0.707*Ib;
Izero  = 0.577*Ir + 0.577*Iy + 0.577*Ib;

active_power = Valpha.*Ialpha + Vbeta.*Ibeta;
reactive_power = Valpha.*Ibeta - Vbeta.*Ialpha;

subplot(3,4,8);
plot(t,active_power,t,reactive_power);
```

```
value_act_p = mean(active_power);
value_act_q = mean(reactive_power);

%First quadrant
if(value_act_p > 0 & value_act_q > 0)
    angle_p = 30;
 end

%Second quadrant
if(value_act_p < 0 & value_act_q > 0)
    angle_p = 120;
 end

%Third quadrant
if(value_act_p < 0 & value_act_q < 0)
    angle_p = 210;
 end

%Fourth quadrant
if(value_act_p > 0 & value_act_q < 0)
    angle_p = 310;
end

value = abs(value_act_p);
temp = (angle_p*pi)/180;
[x,y] = pol2cart(temp,value);
subplot(3,4,12);
compass(x,y,'r-');
```

❖ SOURCE CODE FOR TRACKING HARMONICS

```
function [sys,x0,str,ts] = KalmanParameters(t,x,u,flag,Fs,f,sigman,sigmav,M,lambda)

global T P U B;
T = 1/Fs;
wT = 2*pi*f*T;

h1 = 2*pi*f*T;
h3 = 3*2*pi*f*T;
h5 = 5*2*pi*f*T;
h7 = 7*2*pi*f*T;
h9 = 9*2*pi*f*T;
h11 = 11*2*pi*f*T;
h13 = 13*2*pi*f*T;
h15 = 15*2*pi*f*T;
h17 = 17*2*pi*f*T;
h19 = 19*2*pi*f*T;
h21 = 21*2*pi*f*T;
h23 = 23*2*pi*f*T;
h25 = 25*2*pi*f*T;

P = [cos(h1)    -sin(h1)         0           0           0            0           0           0            0
0           0            0           0           0                0           0           0            0
0           0                0           0            0           0           0           0     ;
     sin(h1)     cos(h1)         0           0           0            0           0           0            0
0           0            0           0           0                0           0           0            0
0           0                0           0            0           0           0           0     ;
     0              0       cos(h3)      -sin(h3)        0            0           0           0            0
0           0            0           0           0                0           0           0            0
0           0                0           0            0           0           0           0     ;
```

```
      0             0       sin(h3)      cos(h3)          0            0           0            0            0
0            0            0           0            0                0           0            0            0
0            0                0           0            0            0           0            0     ;
      0             0            0            0         cos(h5)     -sin(h5)       0            0            0
0            0            0           0            0                0           0            0            0
0            0                0           0            0            0           0            0     ;
      0             0            0            0         sin(h5)     cos(h5)        0            0            0
0            0            0           0            0                0           0            0            0
0            0                0           0            0            0           0            0     ;
      0             0            0            0           0            0       cos(h7)     -sin(h7)          0
0            0            0           0            0                0           0            0            0
0            0                0           0            0            0           0            0     ;
      0             0            0            0           0            0       sin(h7)     cos(h7)           0
0            0            0           0            0                0           0            0            0
0            0                0           0            0            0           0            0     ;
      0             0            0            0           0            0           0            0       cos(h9)
-sin(h9)         0            0           0            0                0           0            0            0
0            0                0           0            0            0           0            0     ;
      0             0            0            0           0            0           0            0       sin(h9)
cos(h9)          0            0           0            0                0           0            0            0
0            0                0           0            0            0           0            0     ;
      0             0            0            0           0            0           0            0            0
0       cos(h11)     -sin(h11)      0            0                0           0            0            0
0            0                0           0            0            0           0            0     ;
      0             0            0            0           0            0           0            0            0
0       sin(h11)     cos(h11)       0            0                0           0            0            0
0            0                0           0            0            0           0            0     ;
      0             0            0            0           0            0           0            0            0
0            0            0       cos(h13)     -sin(h13)            0           0            0            0
0            0                0           0            0            0           0            0     ;
      0             0            0            0           0            0           0            0            0
0            0            0       sin(h13)     cos(h13)             0           0            0            0
0            0                0           0            0            0           0            0     ;
      0             0            0            0           0            0           0            0            0
0            0            0           0            0           cos(h15)      -sin(h15)      0            0
0            0                0           0            0            0           0            0     ;
```

```
     0               0           0            0           0            0            0           0           0
0           0            0           0            0           sin(h15)     cos(h15)       0           0
0           0                0           0            0            0            0           0     ;
     0               0           0            0           0            0            0           0           0
0           0            0           0            0                0            0      cos(h17)      -sin(h17)
0           0                0           0            0            0            0           0     ;
     0               0           0            0           0            0            0           0           0
0           0            0           0            0                0            0      sin(h17)      cos(h17)
0           0                0           0            0            0            0           0     ;
     0               0           0            0           0            0            0           0           0
0           0            0           0            0                0            0           0           0
cos(h19)     -sin(h19)           0            0           0            0            0           0      ;
     0               0           0            0           0            0            0           0           0
0           0            0           0            0                0            0           0           0
sin(h19)     cos(h19)            0            0           0            0            0           0      ;
     0               0           0            0           0            0            0           0           0
0           0            0           0            0                0            0           0           0
0           0            cos(h21)     -sin(h21)       0            0            0           0     ;
     0               0           0            0           0            0            0           0           0
0           0            0           0            0                0            0           0           0
0           0            sin(h21)      cos(h21)       0            0            0           0     ;
     0               0           0            0           0            0            0           0           0
0           0            0           0            0                0            0           0           0
0           0                0           0        cos(h23)     -sin(h23)        0           0     ;
     0               0           0            0           0            0            0           0           0
0           0            0           0            0                0            0           0           0
0           0                0           0        sin(h23)      cos(h23)        0           0     ;
     0               0           0            0           0            0            0           0           0
0           0            0           0            0                0            0           0           0
0           0                0           0            0            0       cos(h25)     -sin(h25);
     0               0           0            0           0            0            0           0           0
0           0            0           0            0                0            0           0           0
0           0                0           0            0            0       sin(h25)     cos(h25)];

 %P = [cos(wT) -sin(wT) 0;sin(wT) cos(wT) 0;0 0 exp(-lambda*T)];
% M = [m^2 0 0;0 m^2 0;0 0 m^2];
```

```
U = [sigmav^2       0       0           0       0       0       0       0       0           0           0
0           0           0       0               0           0           0           0           0
0           0           0           0           0           0    ;
        0       sigmav^2    0           0       0       0       0       0       0           0           0
0           0           0       0               0           0           0           0           0
0           0           0           0           0           0    ;
        0           0     sigmav^2      0       0       0       0       0       0           0           0
0           0           0       0               0           0           0           0           0
0           0           0           0           0           0    ;
        0           0       0       sigmav^2    0       0       0       0       0           0           0
0           0           0       0               0           0           0           0           0
0           0           0           0           0           0    ;
        0           0       0           0   sigmav^2    0       0       0       0           0           0
0           0           0       0               0           0           0           0           0
0           0           0           0           0           0    ;
        0           0       0           0       0   sigmav^2    0       0       0           0           0
0           0           0       0               0           0           0           0           0
0           0           0           0           0           0    ;
        0           0       0           0       0       0   sigmav^2    0       0           0           0
0           0           0       0               0           0           0           0           0
0           0           0           0           0           0    ;
        0           0       0           0       0       0       0   sigmav^2    0           0           0
0           0           0       0               0           0           0           0           0
0           0           0           0           0           0    ;
        0           0       0           0       0       0       0       0   sigmav^2        0           0
0           0           0       0               0           0           0           0           0
0           0           0           0           0           0    ;
        0           0       0           0       0       0       0       0       0       sigmav^2        0
0           0           0       0               0           0           0           0           0
0           0           0           0           0           0    ;
        0           0       0           0       0       0       0       0       0           0
sigmav^2        0           0           0       0               0           0           0           0
0           0           0           0           0           0           0    ;
        0           0       0           0       0       0       0       0       0           0           0
sigmav^2        0           0       0               0           0           0           0           0
0           0           0           0           0           0    ;
```

```
        0           0       0           0       0       0       0        0       0           0           0
0       sigmav^2        0       0               0           0            0           0           0
0           0           0           0           0           0    ;
        0           0       0           0       0       0       0        0       0           0           0
0           0       sigmav^2    0               0           0            0           0           0
0           0           0           0           0           0    ;
        0           0       0           0       0       0       0        0       0           0           0
0           0           0       sigmav^2        0           0            0           0           0
0           0           0           0           0           0    ;
        0           0       0           0       0       0       0        0       0           0           0
0           0           0       0          sigmav^2         0            0           0           0
0           0           0           0           0           0    ;
        0           0       0           0       0       0       0        0       0           0           0
0           0           0       0              0       sigmav^2          0           0           0
0           0           0           0           0           0    ;
        0           0       0           0       0       0       0        0       0           0           0
0           0           0       0              0            0       sigmav^2         0           0
0           0           0           0           0           0    ;
        0           0       0           0       0       0       0        0       0           0           0
0           0           0       0              0            0            0       sigmav^2        0
0           0           0           0           0           0    ;
        0           0       0           0       0       0       0        0       0           0           0
0           0           0       0              0            0            0           0       sigmav^2
0           0           0           0           0           0    ;
        0           0       0           0       0       0       0        0       0           0           0
0           0           0       0              0            0            0           0           0
sigmav^2        0           0           0           0           0    ;
        0           0       0           0       0       0       0        0       0           0           0
0           0           0       0              0            0            0           0           0
0       sigmav^2        0           0           0           0    ;
        0           0       0           0       0       0       0        0       0           0           0
0           0           0       0              0            0            0           0           0
0           0       sigmav^2        0           0           0    ;
        0           0       0           0       0       0       0        0       0           0           0
0           0           0       0              0            0            0           0           0
0           0           0       sigmav^2        0           0    ;
```

```
        0           0       0           0       0       0       0       0       0           0           0
0           0           0       0               0           0           0           0           0
0           0           0           0       sigmav^2        0    ;
        0           0       0           0       0       0       0       0       0           0           0
0           0           0       0               0           0           0           0           0
0           0           0           0           0       sigmav^2];

%U = [sigmav^2 0 0;0 sigmav^2 0;0 0 sigmav^2];
B = [sigman^2];% B = [sigman2 0 0;0 sigman2 0;0 0 sigman2];
switch flag,
    case 0
        [sys,x0,str,ts] = mdlInitializeSizes(Fs,f,sigman,sigmav,M,lambda,T,P,U,B,wT); % Initialization
    case 3
        sys = mdlOutputs(t,x,u); % Calculate outputs
    case { 1, 2, 4, 9 }
        s = []; % Unused flags
    otherwise
        error(['Unhandled flag = ',num2str(flag)]); % Error handling
end;
% End of KalmanParameters function.


function [sys,x0,str,ts] = mdlInitializeSizes(Fs,f,sigman,sigmav,M,lambda,T,P,U,B,wT)
% Call function simsizes to create the sizes structure.
warning off;
sizes = simsizes;
% Load the sizes structure with the initialization information.
sizes.NumContStates= 0;
sizes.NumDiscStates= 0;
sizes.NumOutputs= 0;
sizes.NumInputs= 0;
sizes.DirFeedthrough=1;
sizes.NumSampleTimes=1;
% Load the sys vector with the sizes information.
sys = simsizes(sizes);
%
x0 = []; % No continuous states
%
```

```
str = []; % No state ordering
%
ts = [T 0]; % Inherited sample time
% End of mdlInitializeSizes.
%==============================================================
% Function mdlOutputs performs the calculations.
%==============================================================
function sys = mdlOutputs(t,x,u)
% global T P U B;
sys = [];


function [sys,x0,str,ts] = kfamp(t,x,u,flag,X0,C,M)
global T P U B K;
switch flag,
    %%%%%%%%%%%%%%%%%%
    % Initialization %
    %%%%%%%%%%%%%%%%%%
    case 0,
        [sys,x0,str,ts] = mdlInitializeSizes(X0,C,M,T,P,U,B,K); % Initializatio
        %%%%%%%%%%
        % Update %
        %%%%%%%%%%
    case 2,
        sys = mdlUpdate(t,x,u,C,M);
        %%%%%%%%%%%
        % Outputs %
        %%%%%%%%%%%
    case 3,
        sys = mdlOutputs(t,x,u); % Calculate outputs
    case 9,
        sys = []; % Do nothing
        %%%%%%%%%%%%%%%%%%%%
        % Unexpected flags %
        %%%%%%%%%%%%%%%%%%%%
    otherwise
        error(['Unhandled flag = ',num2str(flag)]); % Error handling
end
```

```
%END of kfamp function

%
%=============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================================
%
function [sys,x0,str,ts] = mdlInitializeSizes(X0,C,M,T,P,U,B,K)
%
% call simsizes for a sizes structure, fill it in and convert it to a
% sizes array.
%
% Note that in this example, the values are hard coded.  This is not a
% recommended practice as the characteristics of the block are typically
% defined by the S-function parameters.
%
% global T P K;
sizes = simsizes;

sizes.NumContStates  = 0;
sizes.NumDiscStates  = 728;                         %783
sizes.NumOutputs     = 3;
sizes.NumInputs      = 1;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;   % at least one sample time is needed

sys = simsizes(sizes);
%
% initialize the initial conditions
%
if isempty(K),
    x0  = [X0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;reshape(M,676,1)];     %16
else
    x0  = [X0;K;reshape(M,676,1)];                  %16
end
```

```
%
% Set str to an empty matrix.
%
str = [];

%
% initialize the array of sample times
%
ts  = [T 0];% sample time: [period, offset]

% END mdlInitializeSizes
%
%=============================================================================
% mdlUpdate
% Handle discrete state updates, sample time hits, and major time step
% requirements.
%=============================================================================
%
function sys = mdlUpdate(t,x,u,C,M)

global P B U K;
Q = P;
s = length(x);
if s>3,
    X = x(1:26,1);                   %
    K = x(27:52,1);                  %
    M = reshape(x(53:728,1),26,26);  %
else
    X = x;
end
PX = P*X;
CPX = C*PX;
K = (M*C')*inv(((C*M*C')+B));
Z = (M-(K*C*M));
M = (P*Z*P')+(Q*U*Q');
sys = [PX+(K*(u-CPX));K;reshape(M,676,1)];        %16
% end mdlUpdate
```

```
%
%=============================================================================
% mdlOutputs
% Return the block outputs.
%=============================================================================
%
function sys = mdlOutputs(t,x,u)

s = length(x);
if s>3,
    sys = x(1:3,1);
else
    sys = x;
end
% end mdlOutputs
```

❖ SOURCE CODE FOR FLICKER MEASUREMENT

```
P0 = 0;
P1 = 0;
P2 = 0;
P3 = 0;
P4 = 0;

angle0 = 0:1:120;
angle1 = 0:1:1200;
angle2 = 0:1:3600;
angle3 = 0:1:12000;
angle4 = 0:1:60000;

k0 = (Flicker_freq * 120)/Sampling_freq;
k1 = (Flicker_freq*1200)/Sampling_freq;
k2 = (Flicker_freq*3600)/Sampling_freq;
k3 = (Flicker_freq*12000)/Sampling_freq;
k4 = (Flicker_freq*60000)/Sampling_freq;

Record_len0 = 0.6;
Record_len1 = 6;
Record_len2 = 18;
Record_len3 = 60;
Record_len4 = 300;

t = 0:0.001:10;
t0 = 0:0.005:0.6;
t1 = 0:0.005:6;
t2 = 0:0.005:18;
t3 = 0:0.005:60;
t4 = 0:0.005:300;

Mag_flk_freq0 = 0.256;
Mag_flk_freq1 = 0.256;
Mag_flk_freq2 = 0.256;
Mag_flk_freq3 = 0.256;
Mag_flk_freq4 = 0.256;

voltage_sg = Input_mag*cos(2*pi*Input_freq*t)+
Flicker_mag*cos(2*pi*Flicker_freq*t)
%plot(t,voltage_sg);
%xlabel('Time (msec)');
%ylabel('Amplitude');
%title('VOLTAGE WAVEFORM');

output_sg_0 = Input_mag*cos(2*pi*Input_freq*t0)+
Flicker_mag*cos(2*pi*Flicker_freq*t0)
%subplot(3,2,1);
%plot(t0,output_sg_0);
%xlabel('Time (msec)');
%ylabel('Amplitude');
```

```
%title('Record Length: 0.6 sec');

output_sg_1 = Input_mag*cos(2*pi*Input_freq*t1)+
Flicker_mag*cos(2*pi*Flicker_freq*t1)
%subplot(3,2,2);
%plot(t1,output_sg_1);
%xlabel('Time (msec)');
%ylabel('Amplitude');
%title('Record Length: 6 sec');

output_sg_2 = Input_mag*cos(2*pi*Input_freq*t2)+
Flicker_mag*cos(2*pi*Flicker_freq*t2)
%subplot(3,2,3);
%plot(t2,output_sg_2);
%xlabel('Time (msec)');
%ylabel('Amplitude');
%title('Record Length: 18 sec');

output_sg_3 = Input_mag*cos(2*pi*Input_freq*t3)+
Flicker_mag*cos(2*pi*Flicker_freq*t3)
%subplot(3,2,4);
%plot(t3,output_sg_3);
%xlabel('Time (msec)');
%ylabel('Amplitude');
%title('Record Length: 60 sec');

output_sg_4 = Input_mag*cos(2*pi*Input_freq*t4)+
Flicker_mag*cos(2*pi*Flicker_freq*t4)
%subplot(3,2,5:6);
%plot(t4,output_sg_4);
%xlabel('Time (msec)');
%ylabel('Amplitude');
%title('Record Length 300 sec');

data0 = (2*pi*k0*angle0)/120;
inv_real = transpose(output_sg_0);
real = cos(data0);
temp1 =  real * inv_real;
imag = sin(data0);
temp2 = imag * inv_real;
fft_sig_0 = 2*sqrt(temp1*temp1 + temp2*temp2)/120;

data1 = (2*pi*k1*angle1)/1200;
inv_real = transpose(output_sg_1);
real = cos(data1);
temp1 =  real * inv_real;
imag = sin(data1);
temp2 = imag * inv_real;
fft_sig_1 = 2*sqrt(temp1*temp1 + temp2*temp2)/1200;

data2 = (2*pi*k2*angle2)/3600;
inv_real = transpose(output_sg_2);
real = cos(data2);
temp1 =  real * inv_real;
imag = sin(data2);
```

```
temp2 = imag * inv_real;
fft_sig_2 = 2*sqrt(temp1*temp1 + temp2*temp2)/3600;

data3 = (2*pi*k3*angle3)/12000;
inv_real = transpose(output_sg_3);
real = cos(data3);
temp1 =  real * inv_real;
imag = sin(data3);
temp2 = imag * inv_real;
fft_sig_3 = 2*sqrt(temp1*temp1 + temp2*temp2)/12000;

data4 = (2*pi*k4*angle4)/60000;
inv_real = transpose(output_sg_4);
real = cos(data4);
temp1 =  real * inv_real;
imag = sin(data4);
temp2 = imag * inv_real;
fft_sig_4 = 2*sqrt(temp1*temp1 + temp2*temp2)/60000;

Ref_value0 = int8(Record_len0/Flicker_freq);
Ref_value1 = int8(Record_len1/Flicker_freq);
Ref_value2 = int8(Record_len2/Flicker_freq);
Ref_value3 = int8(Record_len3/Flicker_freq);
Ref_value4 = int8(Record_len4/Flicker_freq);

     if(fft_sig_0 >= Ref_Mag_8_8)
    P0 = Ref_value0;
     end
     if(fft_sig_1 >= Ref_Mag_8_8)
    P1 = Ref_value1;
     end
     if(fft_sig_2 >= Ref_Mag_8_8)
    P2 = Ref_value2;
     end
     if(fft_sig_3 >= Ref_Mag_8_8)
    P3 = Ref_value3;
     end
     if(fft_sig_4 >= Ref_Mag_8_8)
    P4 = Ref_value4;
     end

    Pst_value = (0.0314*P0 + 0.0525*P1 + 0.0657*P2 + 0.28*P3 +
0.08*P4);

    if(Pst_value > 1)
        Pst = 'YES'
        Flicker = 1;
        disp('FLICKER PRESENT');
    else
        Pst = 'NO'
        Flicker = 0;
        disp('FLICKER NOT PRESENT');
    end
```